April 1970

SCENE ANALYSIS USING REGIONS

by

Claude R. Brice
Claude L. Fennema

Paper to be submitted for publication in
the AI Journal.

Artificial Intelligence Group
Technical Note 17

SRI Projects 7494 and 8259

Report Documentation Page

1. REPORT DATE **APR 1970**	2. REPORT TYPE	3. DATES COVERED **00-00-1970 to 00-00-1970**

4. TITLE AND SUBTITLE **Scene Analysis Using Regions**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SRI International,333 Ravenswood Avenue,Menlo Park,CA,94025**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

One of the vision projects of the Stanford Research Institute Artificial Intelligence Group is described. The method employed uses regions as basic data and progresses by successive partitioning of the picture toward an interpretable "goal partition," which is then explored by a heuristic decision tree. A general structure is discussed and an example problem is shown in detail

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**38**	

SCENE ANALYSIS USING REGIONS

by

Claude R. Brice
Claude L. Fennema
Stanford Research Institute
Menlo Park, California 94025

Abstract

One of the vision projects of the Stanford Research Institute

Artificial Intelligence Group is described. The method employed

uses regions as basic data and progresses by successive partitioning

of the picture toward an interpretable "goal partition," which is

then explored by a heuristic decision tree. A general structure is

discussed and an example problem is shown in detail.

SCENE ANALYSIS USING REGIONS

by

Claude R. Brice
Claude L. Fennema

Stanford Research Institute
Menlo Park, California

I   INTRODUCTION

One important component of the SRI automaton project is a set of
programs that provide the automaton with a means of interpreting visual
data. This automaton, or robot, described in detail by Nilsson (1969)
and Munson (1970), is equipped with touch sensors, a range finder, and
a standard vidicon camera. The output of the camera is transmitted by
microwave to an analog-to-digital converter which produces a 120 X 120
digitized picture with 16 levels of gray.

In the early stages of the robot project a prime concern was the
simple task of navigation, for which the vision programs needed only to
find areas of environment through which the robot could freely move.
More recently, however, interest has turned to the more advanced tasks
of collecting various objects and using tools to manipulate these objects.
For these tasks it has become necessary for the programs to isolate and
to recognize objects.

The classical paper of Roberts (1965) describes a method for recog-
nizing objects by transforming a digitized gray-scale picture into a line
drawing, which he then analyzes using mathematical models and projective
transformations. His gray-scale to line-drawing transformation applied a
local edge finder to the picture to find points through which edges prob-
ably pass, and then fit straight lines to these points. His mathematical

-1-

models were then transformed by projection and rotation to find the model that best explained (or fit) the line drawing.

Roberts' treatment of objects was elegant, and many who followed sought to improve the technique with better edge finders and line detectors. Many of these are described and referenced in the work of A. Rosenfeld (1969); other interesting approaches are described by Duda and Hart (1970) and Feldman (1968).

In 1968 Guzman's thesis was published describing a way to isolate objects from a line drawing by observing the vertex configurations of the regions of the picture. This lead to efforts to find regions as well as their boundaries. The Guzman techniques would then group these regions into objects that can be recognized either by Roberts' model matching or by simpler means.

Marvin Minsky and Seymour Pappert (1967) are perhaps the first persons who seriously investigated the direct transformation of a gray-scale picture to regions, bypassing the edge-finding, line-fitting procedures. The method they describe constructs <u>regions</u> <u>that</u> <u>are</u> <u>the</u> <u>union</u> <u>of</u> <u>squares</u> whose corners have the same or nearly the same gray scale.

The process described in this paper breaks the digitized picture into atomic regions of uniform gray scale. Then a pair of heuristics is used to join these regions in such a way as to obtain regions whose boundaries are determined more by the natural lines of the scene than by the artificial ones introduced by quantization and noise. Next, a simple line-fitting technique is used to approximate the region boundaries by straight lines, and finally the scene analyzer interprets the picture using some simple tests on object groups generated by a Guzman-like procedure (Guzman, 1968).

In Section II, the region-oriented structure is described, together with its associated operators and information-retrieval functions. Section III then describes the preprocessing heuristics and the line-fitting procedure, and Section IV describes the scene analyzer.

II   THE DATA REPRESENTATION AND RELATED OPERATIONS

While it is possible to represent the information in a scene accurately by a digitized gray-scale picture, this representation alone does not put emphasis on the interesting properties of the scene. It would be more convenient to have a description in terms of the "natural" elements of the picture, such as regions and lines. For example, if we had an ideal picture of a cube, a natural description would be in terms of regions corresponding to the quadrilaterals and the lines corresponding to the boundaries of these quadrilaterals. In reality, pictures are far from ideal, and many steps of processing are necessary; however, by using regions and lines to describe the data of each step we will be working in terms of the elements of the picture--the boundaries, regions, and their properties will be reorganized, but they will be represented in a uniform manner. Furthermore, describing pictures in terms of regions provides easy access to some useful global information.

A. Preliminary Definitions

A picture is first stored in the computer as an array P on an n x m grid G. The array P is a function on G, whose value on each point is the gray scale of that point. [More generally, $P(i,j)$ could also include other properties such as range, color, and texture.] Each pair (i,j) is called a picture element, whose gray scale is the value of $P(i,j)$ and whose coordinates are i and j; thus, a picture may be thought of as a set of picture elements (provided we remember the topological structure imposed by the grid G).

In what follows we often refer to neighbors of a point (i,j), by which we mean any of the four points that are nondiagonally adjacent (see Fig. 1). We say that two points p_1 and p_2, belonging to a subset R of G, are connected with respect to the set R, provided that there exists a sequence of points from R, the first of which is p_1 and the last of which is p_2, and such that consecutive pairs are neighbors. With this definition in mind, we then define a region as a set of $R \subseteq G$ in which any pair of points is connected with respect to R. These regions are the basic elements of the picture representation.

A partition of a set X is any collection of sets $\{R_1, R_2, \ldots, R_n\}$ such that the union of the R_k is exactly X and the pairwise intersection of the R_k is nil unless the two sets are identical.

If we define some equivalence relation on the array P--say, as a trivial example, $P(i,j)$ is equivalent to $P(k,\ell)$ if their values are equal--then this in turn induces a natural equivalence relation on G given by: (i,j) is equivalent to (k,ℓ) if and only if $P(i,j)$ is equivalent to $P(k,\ell)$.

Any equivalence relation on G yields a partition of G into equivalence classes. These classes can be further broken down into maximally connected subsets called connected components; we call these homogeneous connected components atomic regions. Using the equivalence relation induced by the equality of the gray scale, the atomic regions are obtained by the first step of our technique. They are the building blocks that, if properly joined, will give us a representation of the picture in a form usable to the interpretation methods.

4

B. Region Representation

The process described in this paper makes extensive use not only of the region boundaries, but also of the local properties of the points they surround. Furthermore, during the analysis, regions are constantly being joined with other regions, and sometimes a single region is split into two. This makes it difficult, if not impossible, to use the region representation schemes of Cook (1967) and Pfaltz et al. (1968), whose main concern is only with the boundaries themselves and the relationship of the regions with the regions with which they share a common boundary.

One may think of the picture grid G as a subgrid of some super grid S. If G is an n $\times$ m grid, then S is $(2n + 1) \times (2m + 1)$, and the points (i,j) of G are those where i and j are both odd (see Fig. 2). The points of the regions here are represented by appropriate points in G and the boundaries of these regions are closed curves made up of horizontal and vertical line segments whose endpoints are points in S that have even coordinates (see Fig. 2). These boundaries are then between the points they separate. The points of the subgrid B of boundary segment endpoints are of some importance later; we call them boundary points.

Representing regions in this manner admits a simple algorithm for finding the atomic regions of a picture. Each point of G (except for edge effects) is compared with the one above it and the one to its right, and, if a difference in gray scale is encountered, the boundary segment is inserted between them (Fig. 3). When each point has been considered the grid is partitioned into regions.

5

C.   Important Operations

At all times the region boundaries are assumed to be oriented in such a way that the region lies to the left of the boundary. This allows a simple means of performing the join operation MERGE. This operation merges two regions into one by adding their boundaries as in Fig. 4.

A second operation, CUT, completes the list of operations that affect the picture partition. This operation splits a region along a straight line (see Fig. 5). A more general operation can be defined to cut along arbitrary curves, but straight lines are sufficient in our current problem domain.

Starting with the original partition into atomic regions, a few simple heuristics are used to guide these operations in their repartitioning of the picture. We next describe these heuristics.

III   INITIAL PROCESSING

The TV pictures (as in Fig. 6) are first digitized (Fig. 7), then are partitioned into homogeneous connected components. As can be seen from Fig. 8, this first partition does not typically permit simple interpretation within the context of the problem domain. Many false boundaries are created by lens distortion, uneven illumination, shadows, reflections, distance from the light source, noise and nonuniformities in the composition of the surfaces of the objects. Furthermore, the straight lines of the pictures are not easily extracted from their quantized representations.

In this section we describe two heuristics that effect a repartition of a picture, and a process for fitting lines to the boundaries.

6

A. <u>Repartitioning</u>

One approach to obtaining the desired regions of the picture is to group points if certain of their properties are not too different. It is difficult, however, to use such a method without producing regions that extend beyond the natural lines of the picture. If, for example, two natural regions are even only locally--not too different across the boundary that separates them--they will be considered as one. The problem becomes, then, one of breaking regions into the natural regions they contain. Then a picture partition will often consist of only one region with many boundaries. This presents problems quite similar to those of analyzing a gradient picture.

The procedure chosen here is to start with the atomic regions and use more global criteria to join them. Because these criteria consider the entire boundaries of two regions to be joined, the regions are not as often erroneously joined.

The assumption used to further process such a picture is that boundaries between regions belonging to the same surface generally are not as "strong" as boundaries between regions from different surfaces. On the other hand, while it is often true that intersections of two different surfaces produce the strong boundary segments, this criterion of boundary strength is not sufficiently reliable. More sophisticated criteria are needed to form significant regions. The heuristics described below guide the merging of regions to yield a new partition of the picture that respects the natural lines of the picture.

7

1. The Phagocyte Heuristic

The first heuristic tends to guide the merging of regions in such a way as to smooth or shorten the resulting boundary. Two regions that differ strongly along their common boundary are never joined; but even if this boundary is weak they are joined only if the resulting boundary does not grow too fast.

More precisely, the strength of each elementary vector of a boundary is defined as the difference between the properties of the picture elements on the right and on the left of this vector--in our case, the absolute value of the difference in gray scale of the two elements. The strength of a curve is the average of the strength of each vector of the curve. We are particularly interested in those parts of boundaries where the strength is small, since these are potentially places where regions can be joined. We define the length W of the weak part of the boundary between two regions as the number of boundary vectors having a strength less than some threshold σ. Then, the phagocyte heuristic is to merge adjacent regions R_i and R_j if $W/PM > \theta_1$, where $PM = Min(P_i, P_j)$, P_i is the perimeter of R_i, P_j is the perimeter of R_j, and θ_1 is a threshold. The threshold σ is hardware dependent, and grows with the number of levels of gray and the dynamic range of the picture. The threshold θ_1 is important. If θ_1 is small, the criterion is weak and many regions may be joined. On the other hand, if θ_1 is large, two regions are joined only if one of the regions almost surrounds the other.

Another way of analyzing this heuristic is to observe the growth of the boundary of the resulting region as new regions are joined. If

8

I is the length of the boundary between R_i and R_j, the perimeter P_R of the union of R_i and R_j,

$$P_R = P_i + P_j - 2I$$

and

$$I/\theta_1 > W/\theta_1 > PM = P_j,$$

say. So

$$P_R < P_i + I\left(\frac{1}{\theta_1} - 2\right) \ .$$

This formula shows how the growth of the perimeter P_R is related to θ_1. Note that the value $\theta_1 = \frac{1}{2}$ is significant. For $\theta_1 > \frac{1}{2}$ the boundary must shrink, and for $\theta_1 < \frac{1}{2}$ it is allowed to grow. In Fig. 9 the regions in (a) would be joined if $\theta_1 > \frac{1}{2}$, but those in (b) would not. If θ_1 is sufficiently small, however, both configurations could be joined.

Theoretically, this criterion is order dependent. For example, three regions of Fig. 10 will be joined differently if we first join R_1 and R_2 that they will if we first join R_2 and R_3 (let $\theta_1 = .51$). However, this effect in our experience has not proved to be very important.

This criterion is recursively applied to the regions of a picture until no two regions of the picture satisfy it. Results of this heuristic on real examples have not been extensively tested, but some typical results are shown in Fig. 11. The "smoothing" character allows the complete joining of parts of the face of the wedge [Fig. 11(b)]. The thresholds here were $\sigma = 2$ and $\theta_1 = .45$.

9

2. Weakness Heuristic

We have seen that the phagocyte heuristic cleans up the pictures considerably, but it is clear that a lot of false separations still exist, and more processing is needed.

The second heuristic, the weakness heuristic, is probably more natural than the first in the sense that it joins regions solely on the basis of the strength of the boundary that separates them.

As before, W is the weak portion of the intersection I of two regions R_1 and R_2. Then we join R_1 and R_2 if $W/I > \theta_2$ (i.e., if the weak part is at least a certain percentage of the intersection). This heuristic is applied as before until no further merges are possible. Fig. 12 shows the results (with $\theta_2 = .75$) of applying this criteriom to the pictures of Fig. 11.

This heuristic is more natural than the first, and one might ask why not just apply it alone. The answer is that it is too local. The results are to wipe out almost all the regions. The phagocyte heuristic must be applied first.

These two heuristics yield a partition that admits a more simple interpretation than that of the original. It is also certainly possible that other heuristics could be applied, using more contextual information. For example, a restrictive condition to guide the first two heuristics could be that merging should not be allowed if it tends to break long straight lines. Better partitions could be obtained at greater computation cost, but it is felt here that further perfection lies at a higher level. The application of the above two heuristics yields a considerably cleaner picture, which can be used for the purpose of scene analysis.

10

B. Line Representation

Even after the final partition is obtained, the boundaries
of the regions are still represented as a list of small unit vectors.
This representation is not easily used for shape analysis. To remedy
this problem, we describe in this section a simple straight-line-fitting
program to represent the boundary in terms of long straight lines.

The literature abounds with methods for fitting straight
lines to curves, but we have chosen a simple sliding mask of fixed width
together with some more global criteria to get an approximate line draw-
ing. This by no means gives a perfect line drawing but globablizes the
information fed to the scene analyzer.

The operation consists of three passes. The first pass has
as input data the low-level data structure of elementary vector curves.
Starting with the lowest vertex (a point where three regions come to-
gether) on the boundary of a region, the algorithm applies a mask be-
tween successive points on the boundary until either a vertex is encoun-
tered, or the mask is situated so at least one intermediate point lies
outside the mask (see Fig. 13). When a vertex is encountered, the line
approximation is from the starting point to the vertex. Otherwise, the
endpoints of the line approximation are the starting point and the last
point for which the mask covered the boundary. In either case, the pro-
cedure continues from the new endpoint. The second pass does the same
thing on the output of the first with a more generous mask.

The third pass is similar, but the mask is still larger and
vertices are ignored. This technique is somewhat crude, and Fig. 14
shows that the resulting line drawing is not perfect, but already there
is enough information for the scene analyzer, which we describe next.

11

IV   THE SCENE ANALYZER

This scene analyzer, although simple in form, completes the
present system. Its goal is to interpret the data in terms of the
following object classes:

    (1)   Wedges

    (2)   Cubes

    (3)   Wall

    (4)   Floor.

This problem is simple, but the problem is one of working with im-
perfect data--we deal here with missing lines, broken lines, and
occlusion.

The well known work of Roberts (1965) shows well what can be
done with projective transformation; but we have decided here to ex-
plore the use of clues combined with a two-dimensional description
of the objects, a method that humans use heavily (Gibson, 1966; Yarbus,
1967).

The goal, as we said before, of the scene analyzer is to give an
interpretation of the picture data. Such an interpretation is, say a list

      ((OB1 FLOOR) (OB2 WALL)(OB3 WEDGE))   ,

where each OBN is a collection or group of interpreted regions. By this
we mean that each region in this group is labeled as one of

      (QUAD, TRIANGLE, PART OF FLOOR, PART OF WALL)   .

For each example in the above we might well have

      OB1 =((R17 PART OF FLOOR) (R135 PART OF FLOOR))

      OB2 =((R22 PART OF WALL))

      OB3 = ((R1 QUAD) (R2 TRIANGLE))   .

Basically, the scene analyzer extracts easily recognized regions first (here the floor and wall), then groups the regions, using a simplified Guzman technique, into "object," and then tries to recognize the faces of the objects. If nothing fits, it proposes lines, regroups regions, and begins again.

A. Extracting Easily Recognized Objects

We begin by extracting the floor and walls by using several strong hints provided by the nature of this particular environment.

(1) Floor and wall are separated by a black baseboard of known height.

(2) Floor and walls are light in intensity.

(3) Wall is high in the picture.

(4) Floor is low on the picture.

In many cases this is sufficient information to readily extract floor and walls--expecially if any of the baseboard is visible. If none of these clues are available, the program has to go into a default condition and try to recognize some object in the scene.

B. Grouping Regions

The remaining regions are grouped by a simple Guzman-like technique. Two regions are linked at a vertex if they follow one another going clockwise around that vertex and if both have angles less than 90° at that vertex. We exclude vertices where one angle = 180°, and we do not treat objects that are occluded enough to be divided. These region groups, then, are tentative objects, and a figure recognition program tries to interpret the regions of this object.

13

C. Region Interpretation

The first region of an object is examined to see if it:

(1) Has exactly four sides of reasonable length or

(2) Has exactly three sides of reasonable length.

If it satisfies one of these two criteria it is labeled appropriately
QUAD or TRIANGLE. Otherwise we analyze the region boundary to find
clues that might suggest missing information.

Clues are numerous, but essentially come in two types:

(1) The region boundary touches the edge of the picture
    frame, or some object that is not the floor or wall.

(2) The region boundary contains some syntactic informa-
    tion such as colinear segments.

Clues of the first type indicate that this object may be occluded and
that extending the broken lines into the occluding body may provide the
needed information. This is illustrated by the wedge in the background
of Fig. 14(c). Here, extending the lines beyond the frame of the picture
resolves the interpretation question--this is indeed a triangle.

Clues of type 2 indicate that there are missing lines that
may be proposed in a way suggested by Minsky (1967) if low-level infor-
mation is present or simply inserted if the evidence is strong enough.
We often choose the latter approach, since the low-level information is
sometimes absent.

D. Object Recognition

The object recognition program in its present state is quite
unsophisticated. Its task is to control the grouping and region inter-
pretation.

14

First, the regions remaining after the floor and wall have been extracted are grouped into tenative objects, and a region interpretation is attempted on the regions of the first of these objects. If all the regions of that object are interpretable, the object is marked as recognized and is given an interpretation. This object interpretation is simple:

Wedge = TRIANGLE or TRIANGLE + QUAD

Cube = QUAD + QUAD + QUAD

Unknown = ANY OTHER COMBINATION .

The unknown class has been invented to take care of ambiguous cases. Thus, only two QUADS may be visible, and it is impossible by any means to say whether it is a cube or a wedge with triangular side up.

The region interpreter is not always successful, however. As in the case of Fig. 14(a), line extension is not enough and new regions are formed by inserting the missing lines. In this case, the object recognizer makes the appropriate changes, regroups the regions, and begins again.

E. Example Runs

Fig. 15(a) and 15(b) show the results of the program on two simple scenes--one containing only a cube, the other containing only a wedge. It is worth noting that in 15(a) the missing lines [see Fig. 14(a)] have been entered exactly as described above.

Fig. 16 summarizes the results of the program on the scene of Fig. 6(c). The floor baseboard and wall provide no problems, and the regrouping gives us an easily recognized cube. One of the groups [see Fig. 16(h)] contains only one QUAD that did not satisfy any of the object criteria, and this was left as object unknown. It was, in fact, one reflection on the floor of the baseboard.

15

Fig. 16(i) shows the beginning of the wedge recognition. This triangle was occluded in the picture frame, as well as the cube, but recognition was accomplished by extending the lines above the picture frame. Such information is less reliable than the recognition of unoccluded objects, so the occlusion is remembered.

V   CONCLUSION

The advantages of a uniform data type are perhaps obvious. At any step of processing, information can be extracted in the same way and the same operations can be used throughout. In the above it suffices to have the operations MERGE and CUT, and the local data can always be extracted in the same way--lines, gray scale, neighbors, area, perimeter, etc. Our data type was regions--the original picture was broken into regions of an atomic nature, and this type was kept throughout.

The choice of regions as a data type has proven quite successful. The global information available has led to the two heuristics phagocyte and weakness, which apply global information at the very outset. They use information about the whole region to determine what happens locally. As has been the conjecture of many for some time, global information proves to be quite fruitful when compared with the blind application of local operators. (The application of a local criterion too often blindly destroys needed information.)

The construction of an entire system has permitted the experimentation with various heuristics for repartitioning boundary descriptions, etc. The output of the phagocyte and weakness heuristics has been quite impressive and seems to be relatively independent of context. The results have been to give "clean" pictures in various situations.

16

One of the weaker links in the analysis chain is the line-fitting technique. It is not as faithful as it possibly could be. Neighboring regions are often separated, as in Fig. 14, and T connections not even as faithfully represented as in the original data. A new technique might make better use of the local and global information available.

The scene analyzer described here is very simple and is one of the prime targets for improvement. The next scene analyzer will have to include a new class of objects called "doors" and will make a richer use of the robot's world model to reduce unnecessary analysis of "known" areas. The use of clues together with such a model can make analysis easier.

This new scene analyzer will incorporate the ideas of our co-workers Duda and Hart (1970), which provide several useful tests derived from projective geometry, and the ideas presented here.

REFERENCES

Cook, B. G. A computer representation of plane region boundaries. _Australian Computer Journal_ (November, 1967), 44-50.

Duda, R. O. and Hart, P. E. Experiments in scene analysis. _First National Symposium on Industrial Robots_ Chicago (April 2-3, 1970).

Feldman, J. A. et al. The Stanford hand-eye project. _Proc. of the International Joint Conf. on Artificial Intelligence._ Washington, D.C. (May 1969).

Gibson, J. _The Senses Considered as Perceptual Systems_, Houghton and Mifflin, Boston (1966).

Guzman, A. Decomposition of scenes into bodies. _AFIPS Conference Proc. 33, Part One._ (December, 1968), 291-304.

Minsky, M. A. and S. Pappert, Project MAC Progress Report IV, MIT Press, Cambridge, Mass. (1967).

Munson, J. H. The SRI intelligent automaton program, to be published in _the Proceedings of the First Symposium on Industrial Robots._ Chicago (April 1970).

Nilsson, N. J. A mobile automaton: an application of artificial intelligence techniques. _Proc. International Joint Conf. on Artificial Intelligence._ Washington, D.C. (May, 1969), 509-520.

Pfaltz, J. L., Snively, J. W., Jr., and Rosenfeld, A. _Pictorial Pattern Recognition_, Thompson Book Co., Washington D.C. (1968), 353-371.

Roberts, L. G. _Machine Perception of Three Dimensional Solids_, "Optical and electro-optical information processing," Tippett et al. , eds., MIT Press, Cambridge, Mass. (1965).

Rosenfeld, A. Picture Processing by Computer Academic Press, New York, New York (1969).

Yarbus, A. L. Eye Movement and Vision, Plenum Press, New York, New York (1967).
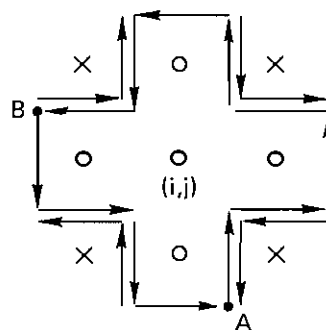
TA-710522-94R

FIGURE 1   A REGION MADE UP OF
            GRIDPOINT (i, j) AND
            ITS FOUR NEIGHBORS

TA-8259-9R
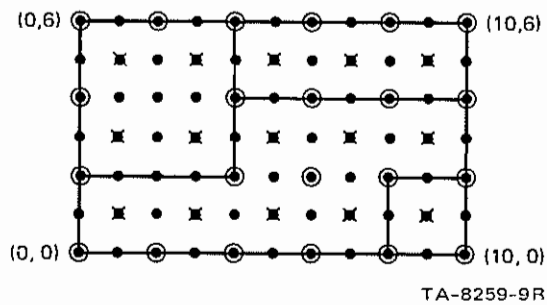
FIGURE 2   FOUR REGIONS REPRESENTED IN THE SUPER
GRID S.   The region points are worked with X,
and the boundary points with O.

TA-8259-10R
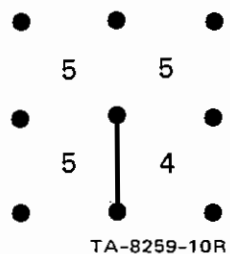
FIGURE 3 INSERTION OF BOUNDARY SEGMENT.
When the lower-left-hand point (each
point of G is replaced by a number
representing its gray scale) is compared
with the point to its right, a difference
is encountered, so a boundary segment
is inserted. A similar segment will be
inserted when the "4" is compared with
the point above it.
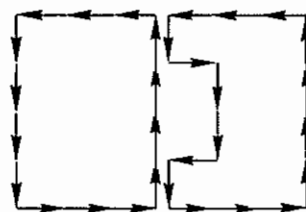
(a)
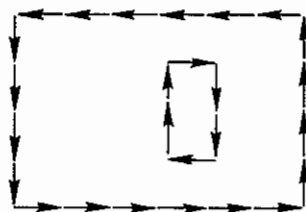
(b)

TA-710522-95R

FIGURE 4   ADDITION OF THE BOUNDARIES OF
TWO REGIONS, AS SHOWN IN (a),
RESULTS IN CANCELLATION OF
THE COMMON PART, AS SHOWN
IN (b)
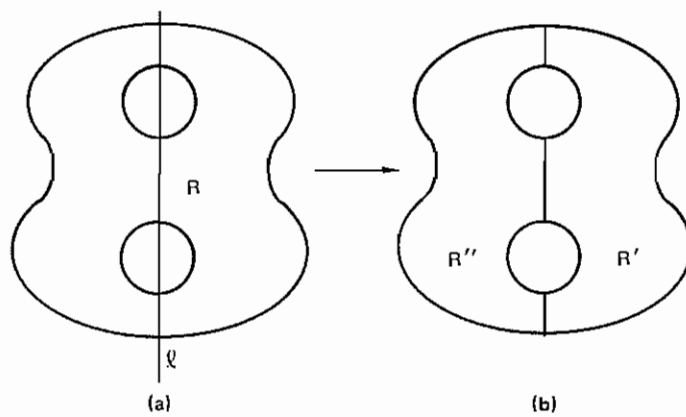
TA-8259-11R

FIGURE 5   THE OPERATOR CUT DIVIDES THE
REGION R INTO TWO PARTS R'
AND R" ALONG THE LINE ℓ.
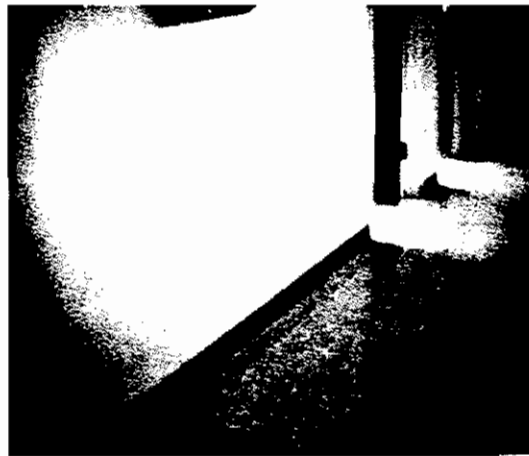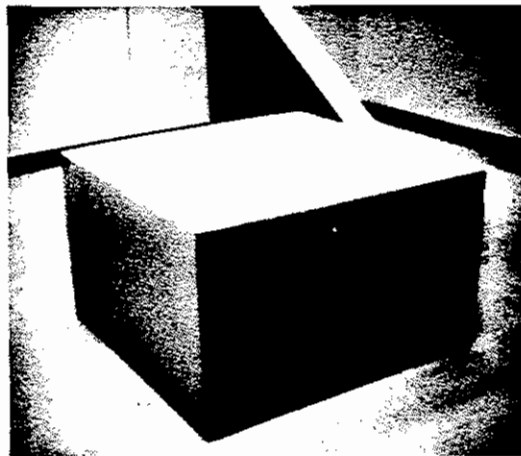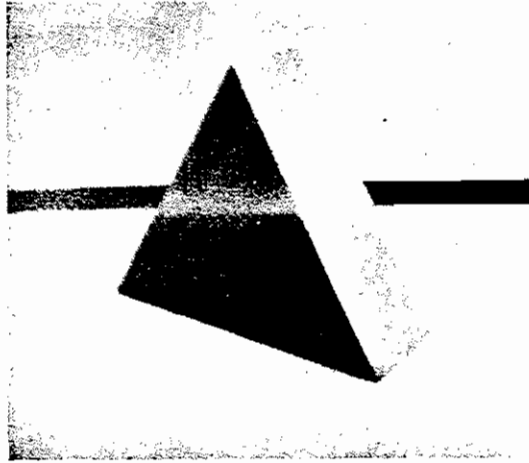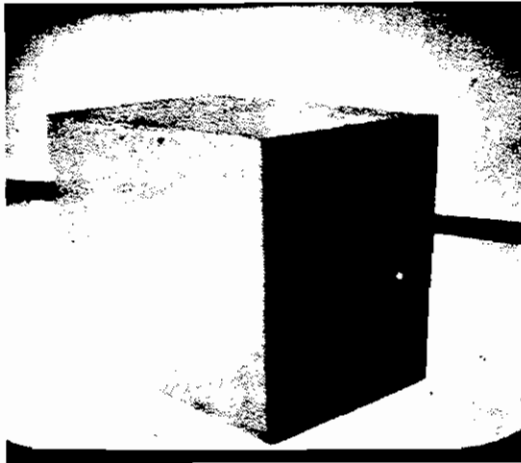
TA-8259-15

FIGURE 6   ORIGINAL OPTICAL PHOTOGRAPHS OF THE EXAMPLES USED IN THIS PAPER
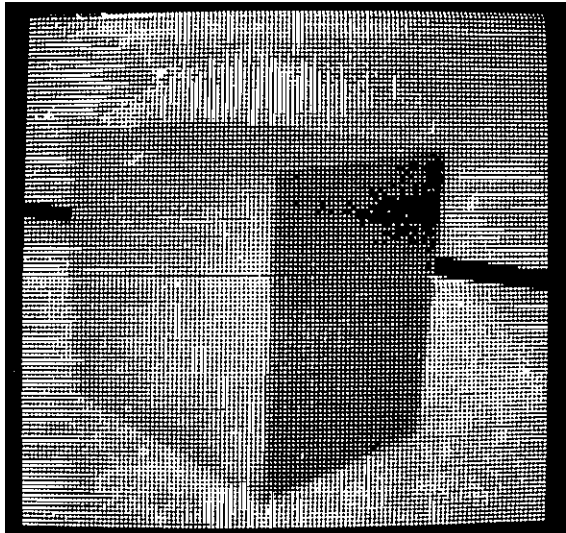
(a)
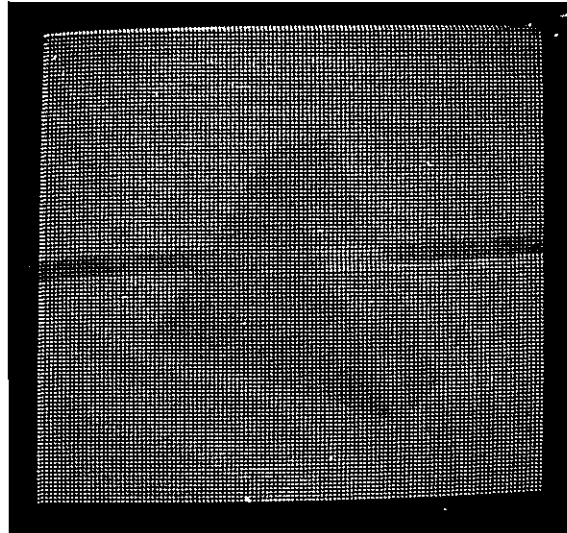
(b)
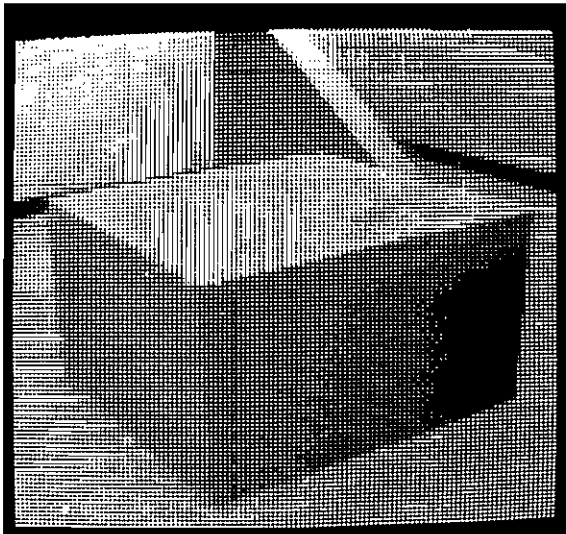
(c)
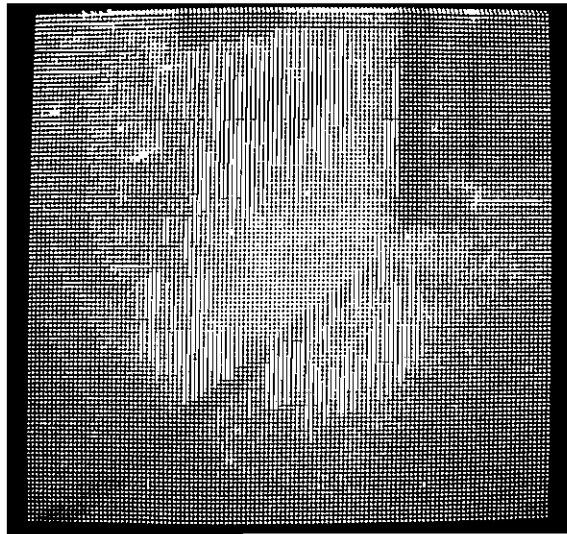
(d)

TA-710522-101R

FIGURE 7   A DISPLAY RENDERING OF THE DIGITIZED GRAY SCALE PICTURES
USED AS EXAMPLES (120 X 120 resolution)
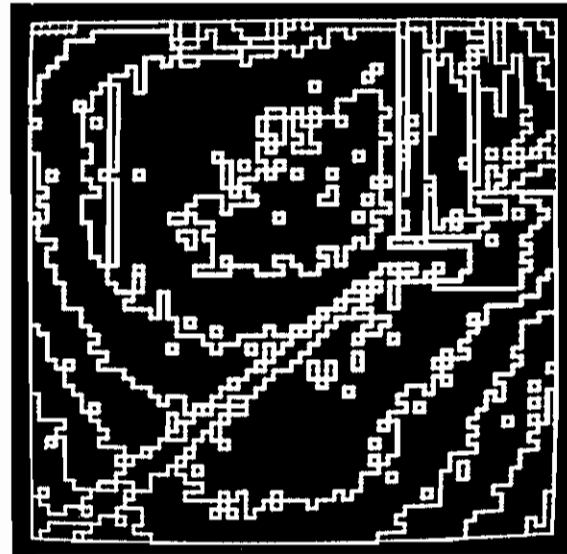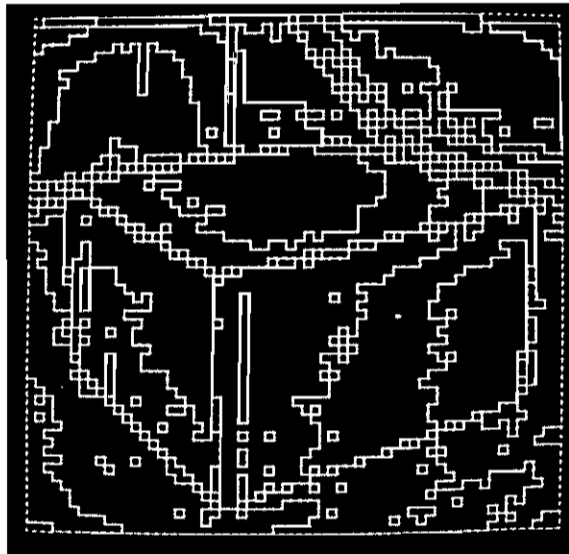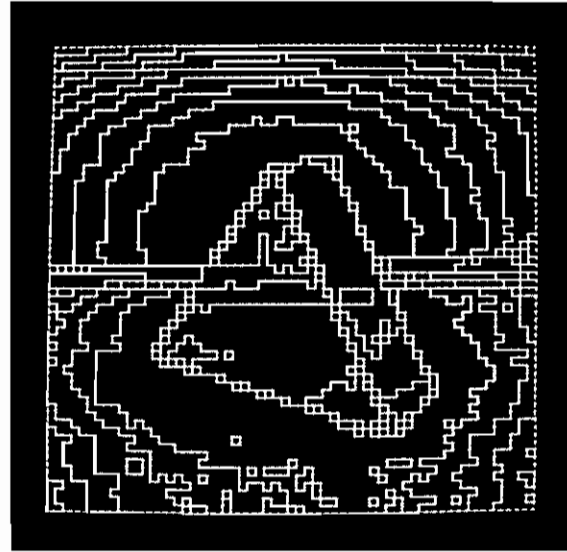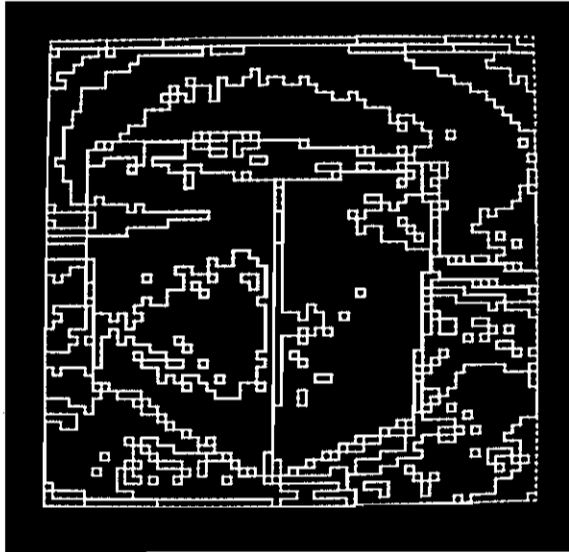
TA-710522-102

FIGURE 8   THE INITIAL PARTITION INTO ATOMIC REGIONS OF THE GRAY SCALE
PICTURES OF FIGURE 6 (60 x 60 resolution)

(a)

(b)
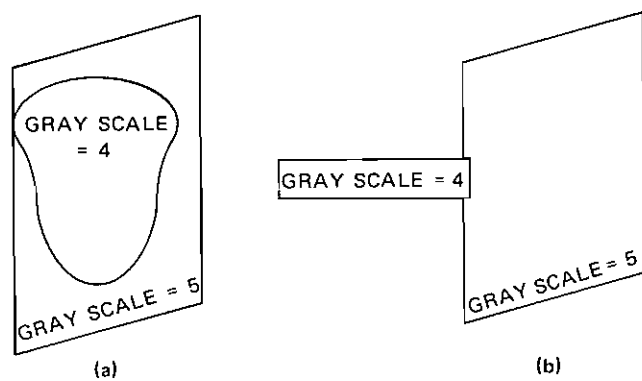
TA-8259-12

FIGURE 9   THE PHAGOCYTE HEURISTIC WILL FOR
θ_1 = 1/2 JOIN THE REGIONS OF (a)
BUT NOT THOSE OF (b).
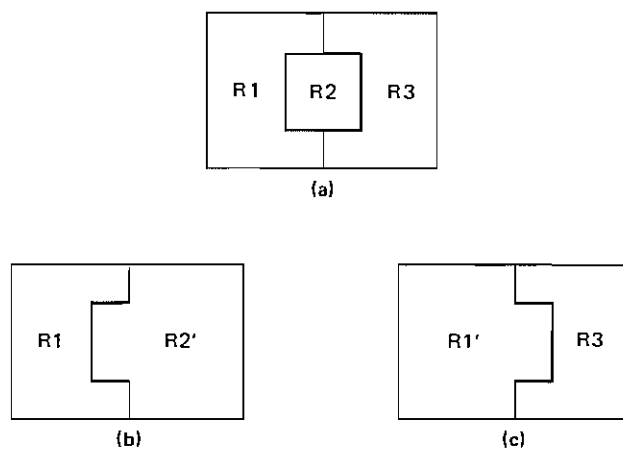
TA-8259-13

FIGURE 10   THE RESULTS OF THE PHAGOCYTE
ALGORITHM ARE ORDER DEPENDENT.
If θ_1 = 1/2, then there are two possible
outcomes to joining R1, R2, and R3,
depending on whether we first join R2 to
R1 or to R3.
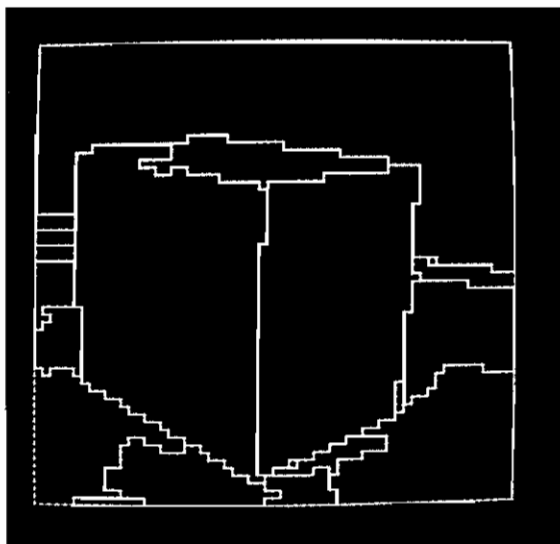
(a)
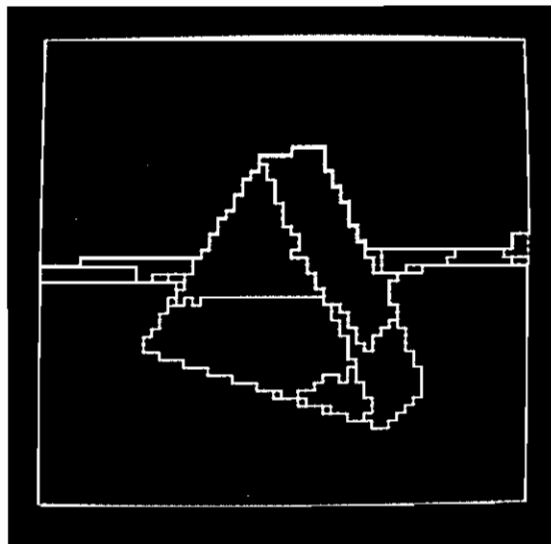
(b)
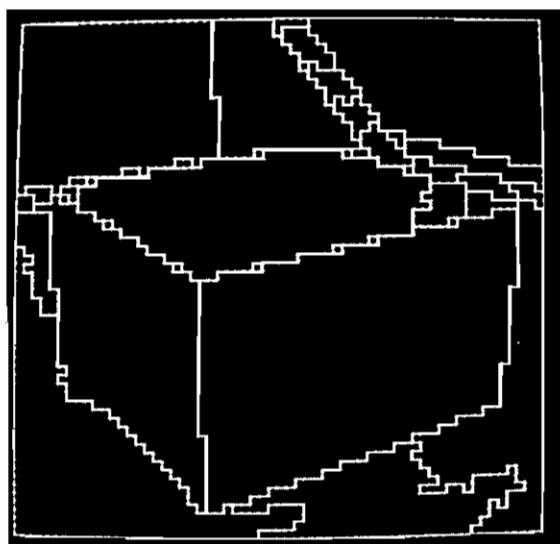
(c)
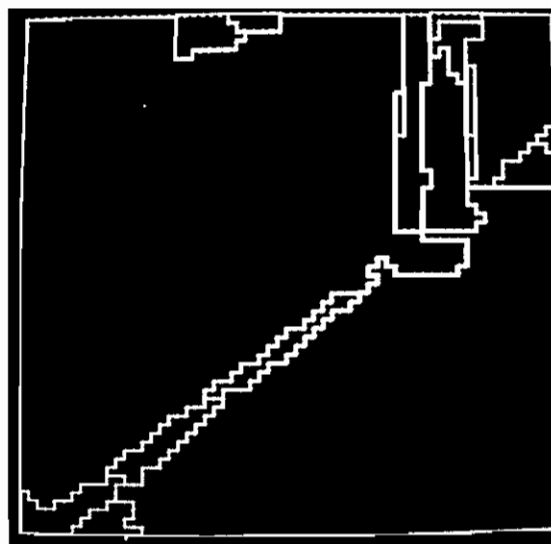
(d)

TA-710522-103R

FIGURE 11   RESULTS OF THE PHAGOCYTE HEURISTIC APPLIED TO THE PARTITIONED
PICTURES   (60 X 60 resolution)
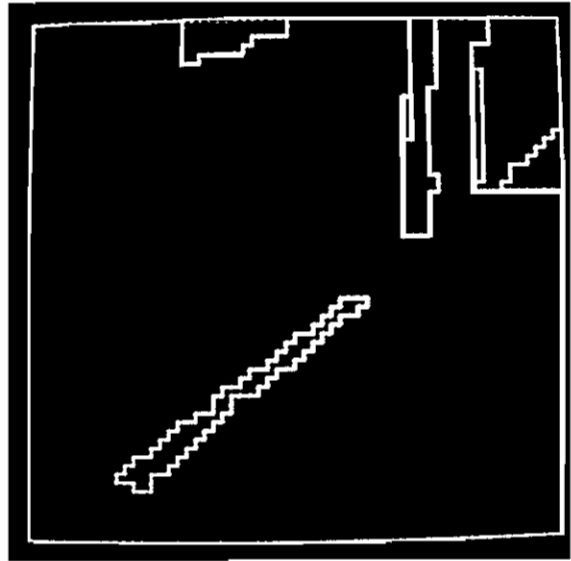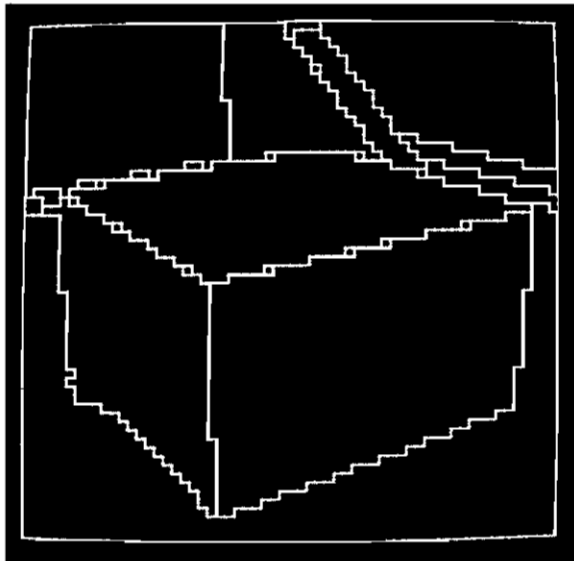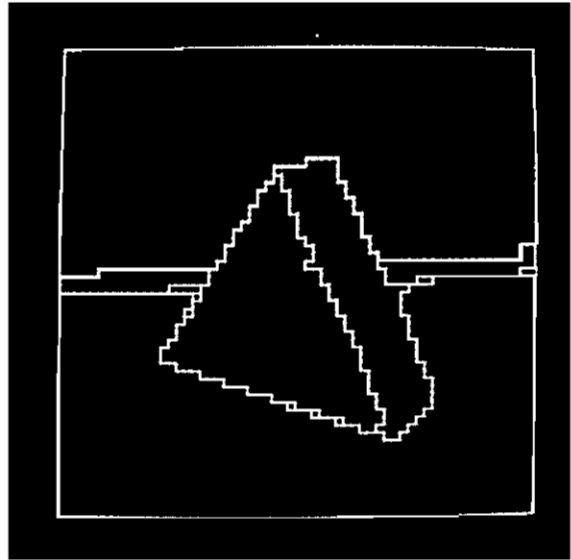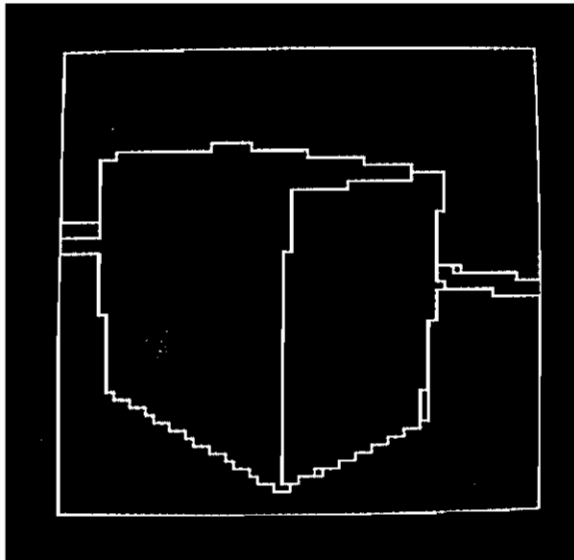
TA-710522-104

FIGURE 12   WEAKNESS HEURISTIC APPLIED TO RESULTS OF THE PHAGOCYTE
HEURISTIC (60 X 60 resolution)
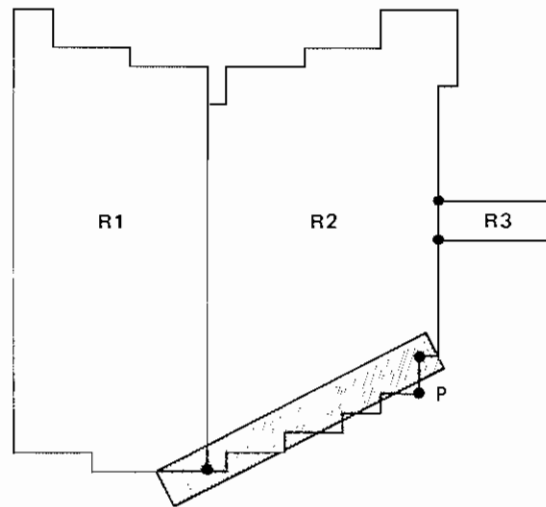
(a)   POINT P LIES OUTSIDE THE MASK
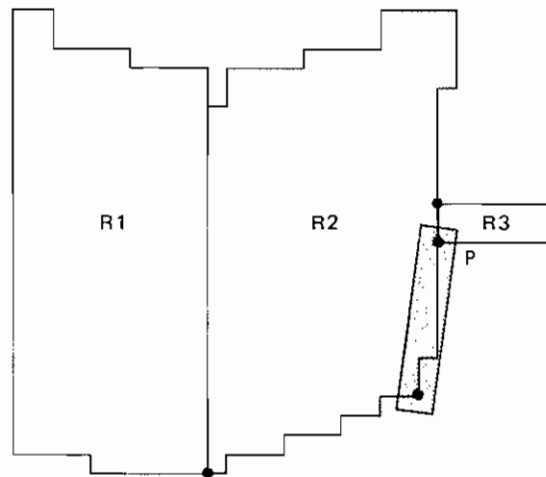
(b)

(b)   VERTEX ENCOUNTERED AT P

TA-8259-14

FIGURE 13   THE USE OF A SLIDING MASK
TO GET AN APPROXIMATE
LINE DRAWING, BY USING
STRAIGHT LINES TO
REPRESENT CURVES

(a)

(b)

(c)
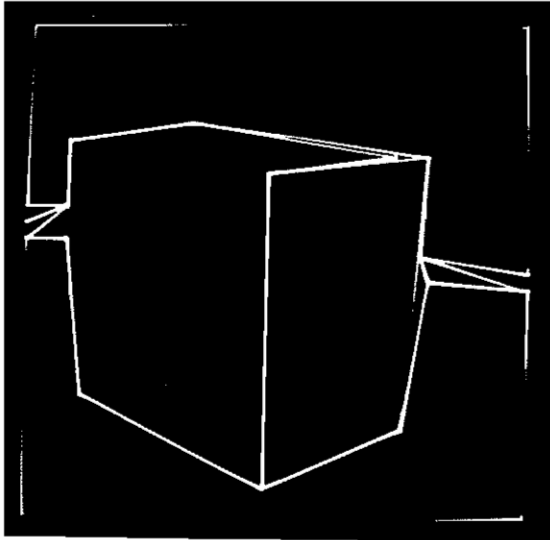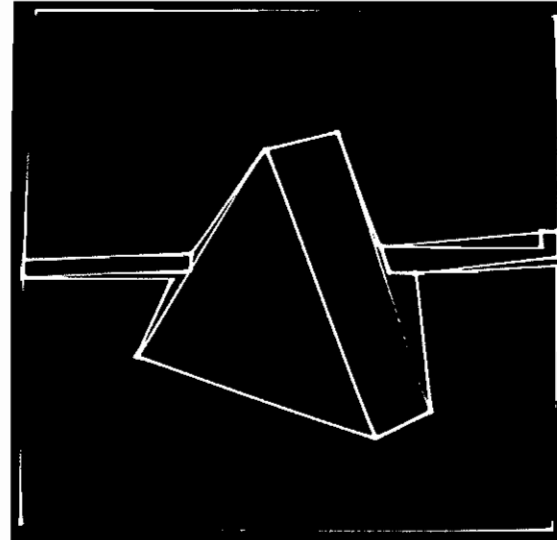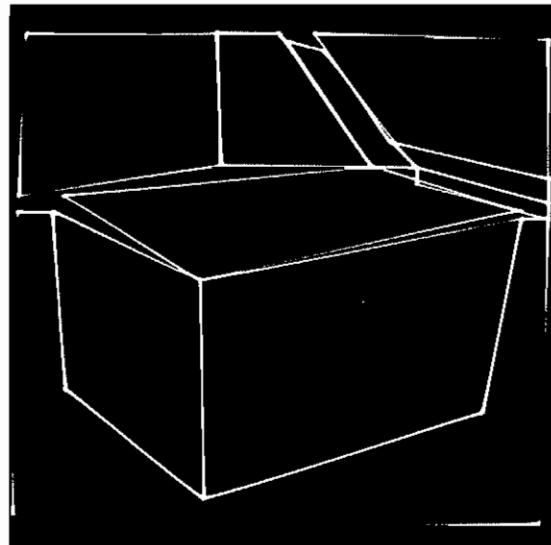
FIGURE 14   RESULTS OF LINE-FITTING TECHNIQUE — HALL OMITTED DUE TO
LACK OF OBJECTS (60 X 60 resolution)
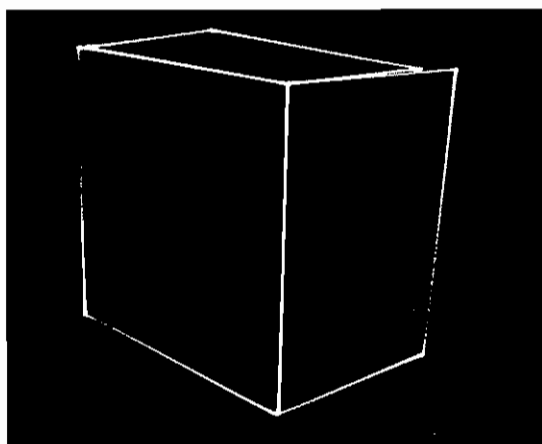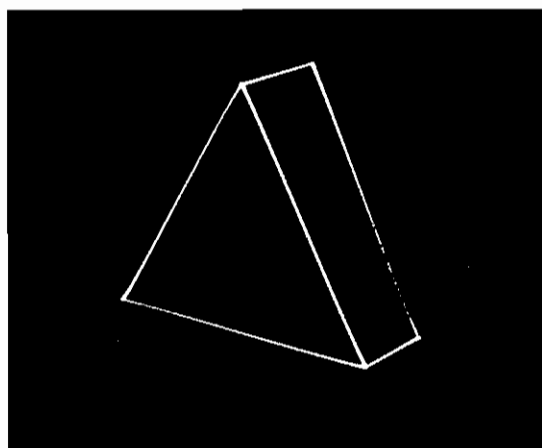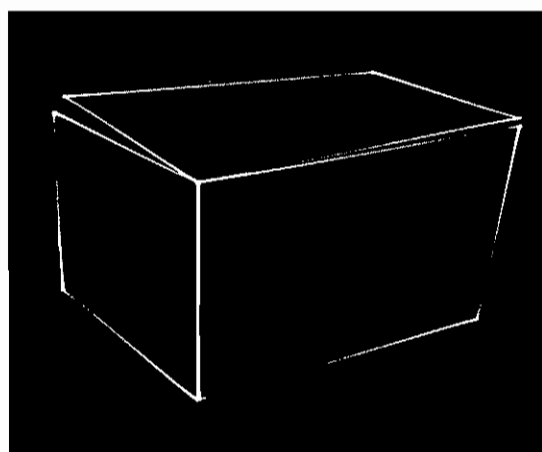
TA-710522-107R

FIGURE 15 OBJECTS ISOLATED — HALL OMITTED DUE TO
LACK OF OBJECTS (60 × 60 resolution)
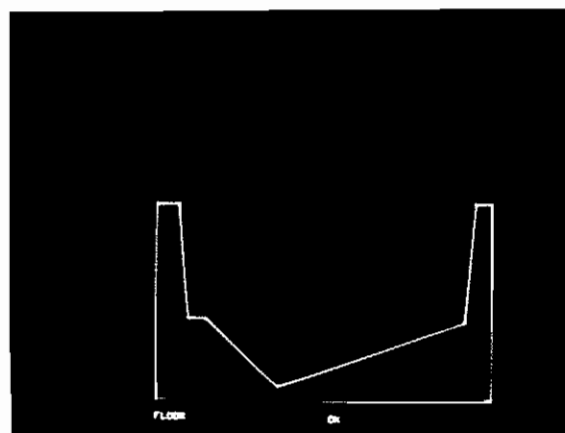
FLOOR   OK
(a)
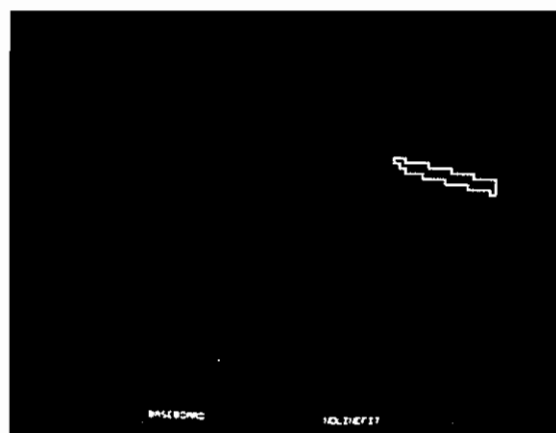
BASEBOARD   NOLINEFIT
(b)
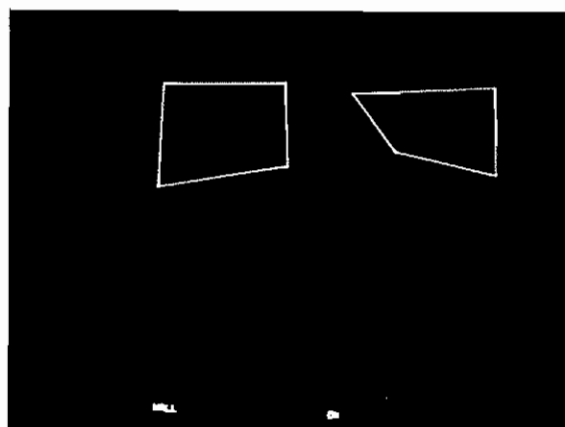
WALL   OK
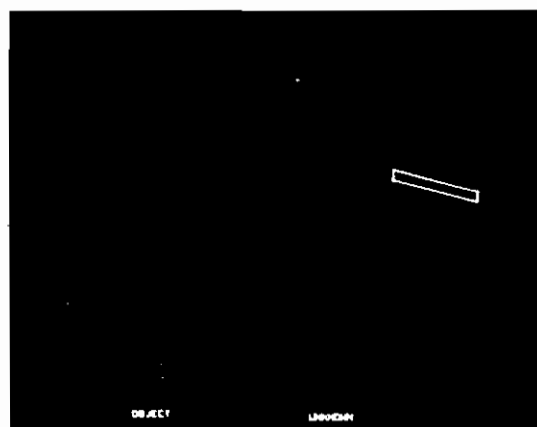(c)
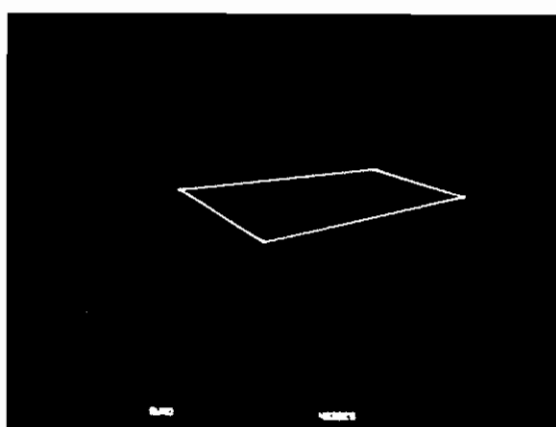
QUAD   4SIDES
(d)
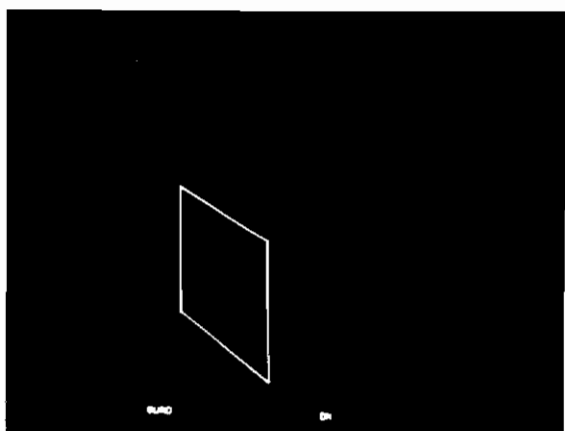
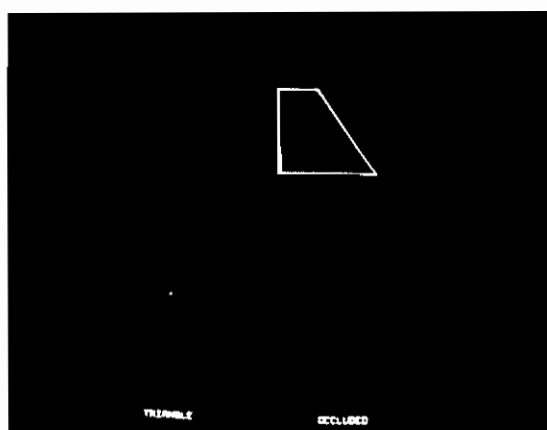QUAD   OK
(e)
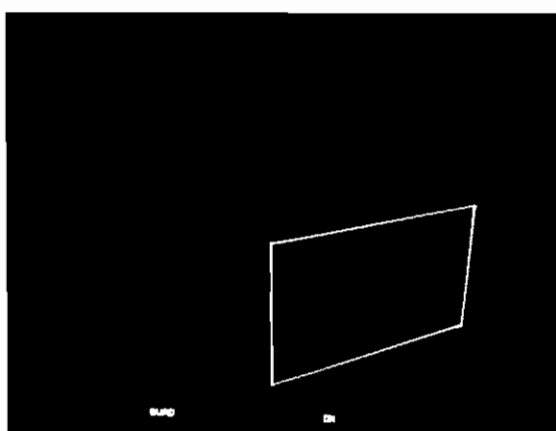
QUAD   OK
(f)

TA—710522—113

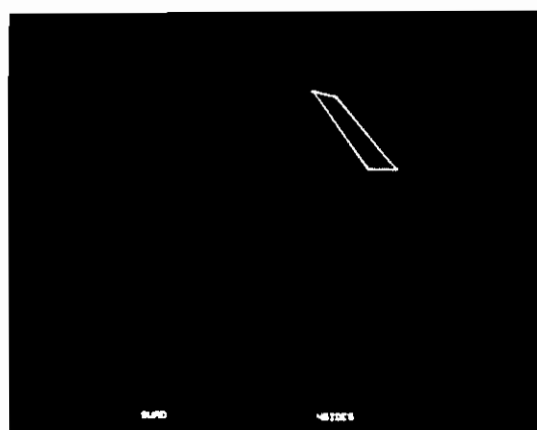FIGURE 16   A SAMPLE RUN OF THE SCENE ANALYSIS PROGRAM SHOWING
INTERMEDIATE STEPS
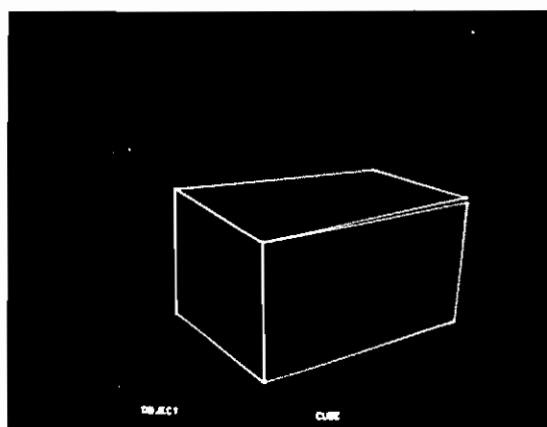
OBJECT   CUBE
(g)

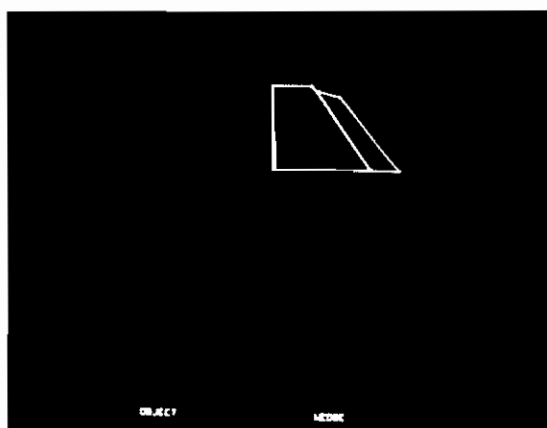OBJECT   UNKNOWN
(h)

TRIANGLE   OCCLUDED
(i)

QUAD   4SIDES
(j)

OBJECT   WEDGE
(k)

(l)

TA-710522-114R

FIGURE 16   Concluded